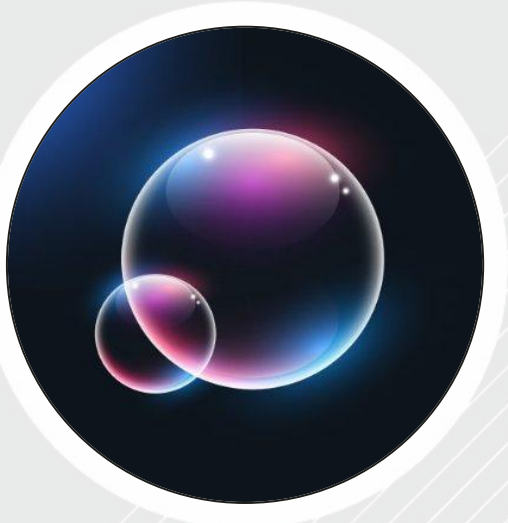
SPYWOLF

Security Audit Report

(TESTNET)



Completed on

December 14, 2022

OVERVIEW

This audit has been prepared for **Bubble DeFi** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

BUBBLE DEFI

PROJECT DESCRIPTION

According to their whitepaper:

Release Date: Presale starts in December, 2022

Category:

CONTRACT 1 INFO

Token Name
BubbleToken

Symbol
$BUB

Contract Address
0xBF96727212E3c7070D9A7db6004bC511031FF9c5

Network
Goerli **TESTNET**

Language
Solidity

Deployment Date
Dec 10, 2022

Verified?
Yes

Total Supply
100,000,000

Status
Not launched

TAXES

Buy Tax
3%

Sell Tax
3%

*Taxes can be changed in future

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of December 14, 2022)

Liquidity

Not added yet

Burn

No burnt tokens

Status:

Not Launched!

MaxTxAmount	DEX
2,000,000	Uniswap

LP Address(es)

Liquidity not added yet

TOKEN TRANSFERS STATS

Transfer Count	TESTNET
Uniq Senders	TESTNET
Uniq Receivers	TESTNET
Total Amount	TESTNET
Median Transfer Amount	TESTNET
Average Transfer Amount	TESTNET
First transfer date	TESTNET
Last transfer date	TESTNET
Days token transferred	TESTNET

SMART CONTRACT STATS

Calls Count	TESTNET
External calls	TESTNET
Internal calls	TESTNET
Transactions count	TESTNET
Uniq Callers	TESTNET
Days contract called	TESTNET
Last transaction time	TESTNET
Created	TESTNET
Create TX	TESTNET
Creator	TESTNET

FEATURED WALLETS

Owner address	0x4b371A173cE974059F43D8219Cfc1972187822a8
LP wallet	0x000000000000000000000000000000000000adD
LP address	**TESTNET**

TOP 3 UNLOCKED WALLETS

1 N/A TESTNET

2 N/A TESTNET

3 N/A TESTNET

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

If minTokenToSwap is set to 0 and contract's token balance is 0, contract will halt on sell and selling will flail.

```
function setMinTokenToSwap(uint256 _amount) external onlyOwner {
    minTokenToSwap = _amount;
}

function distributeAndLiquify(address from, address to) private {
.....................
uint256 contractTokenBalance = balanceOf(address(this));

bool shouldSell = contractTokenBalance >= minTokenToSwap;
.....................
}
```

- Recommendation:
  - Ensure that minTokenToSwap variable is always above zero.

FOUND THREATS

⚠ Medium Risk

Owner can blacklist address making it impossible to sell.

```
function addOrRemoveBots(address[] memory accounts, bool value)
    external
    onlyOwner
{
    for (uint256 i; i < accounts.length; i++) {
        require(
            accounts[i] != address(dexRouter) && !dexPair[accounts[i]],
            "$BUB: cannot blacklist Dex"
        );
        isBot[accounts[i]] = value;
    }
}
```

Owner can set max transaction limit but cannot lower it than 100 $BUB tokens.
Market value of 100 $BUB tokens can be lower than the fees associated with the transaction

```
function setMaxTxnLimit(uint256 _amount) external onlyOwner {
    require(_amount >= 100e9, "$BUB: should be greater than 100 $BUB");
    maxTxnLimit = _amount;
}
```

- Recommendation:
  - Considered as good max transaction limitation is the amount to be not lower than 0.1% of total supply.

FOUND THREATS

⚠ Medium Risk

Owner can set buy/sell fees up to 30%.
Combined buy+sell = 60%.

```
function setBuyFeePercent(uint256 _devFee, uint256 _lpFee)
    external
    onlyOwner
{
    devFeeOnBuying = _devFee;
    liquidityFeeOnBuying = _lpFee;
    require(
        _devFee.add(_lpFee) <= percentDivider.mul(3).div(10),
        "$BUB: can't be more than 30%"
    );
}

function setSellFeePercent(uint256 _devFee, uint256 _lpFee)
    external
    onlyOwner
{
    devFeeOnSelling = _devFee;
    liquidityFeeOnSelling = _lpFee;
    require(
        _devFee.add(_lpFee) <= percentDivider.mul(3).div(10),
        "$BUB: can't be more than 30%"
    );
}
```

- Recommendation:
  - Considered as good tax deduction practice is buy and sell fees combined not to exceed 25%.

Informational

Owner can include/exclude address from fees, max transaction limit and max wallet limit.
If dex pair address is included in max wallet and max wallet limit is too low, selling will fail.

```
function includeOrExcludeFromFee(address account, bool value)
    external
    onlyOwner
{
    isExcludedFromFee[account] = value;
}

function includeOrExcludeFromMaxTxn(address account, bool value)
    external
    onlyOwner
{
    isExcludedFromMaxTxn[account] = value;
}

function includeOrExcludeFromMaxHolding(address account, bool value)
    external
    onlyOwner
{
    isExcludedFromMaxHolding[account] = value;
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

BUBBLE DEFI

GOOD PRACTICES FOUND

- ✔ The owner cannot mint new tokens after deployment
- ✔ The smart contract utilizes "SafeMath" to prevent overflows

CONTRACT 2 INFO

Token Name
PresaleBub

Symbol
N/A

Contract Address
0x37df1e984F3b570D1eABfF4404EA141990c9A6bE

Network
Goerli **TESTNET**

Language
Solidity

Deployment Date
Dec 10, 2022

Verified?
Yes

Total Supply
N/A

Status
Deployed

TAXES

Buy Tax
none

Sell Tax
none

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

FEATURED WALLETS

Owner address	0x4b371A173cE974059F43D8219Cfc1972187822a8
LP address	**Presale contract**

TOP 3 UNLOCKED WALLETS

1 N/A N/A

2 N/A N/A

3 N/A N/A

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ Medium Risk

Owner can enable/disable tokens claim status, making it impossible to claim bought tokens.

```
function setClaim(bool _value) external onlyOwner {
    claimEnable = _value;
}

function claimToken() public {
require(claimEnable,"$BUB: wait for enable claim");
.........
}
```

Informational

Owner can change presale tokens price.

```
function changePrice(uint256 _price) external onlyOwner {
    tokenPerEth = _price;
}
```

Owner can withdraw any tokens from the contract.

```
function changeToken(address _token) external onlyOwner{
    token = IBEP20(_token);
}

function transferFunds(uint256 _value) external onlyOwner {
    owner.transfer(_value);
}

function transferTokens(uint256 _value) external onlyOwner {
    token.transfer(owner, _value);
}
```

Owner can withdraw any tokens from the contract.

```
function setPreSaleTime(uint256 _startTime, uint256 _endTime)
    external
    onlyOwner
{
    preSaleStartTime = _startTime;
    preSaleEndTime = _endTime;
}
```

Informational

Owner can change presale settings - min buy amount, max buy amount, hard cap, total presale supply (totalSupply is for informational purpose only).

```
function setPreSaletLimits(uint256 _minAmount, uint256 _maxAmount,
uint256 _total, uint256 _cap) external onlyOwner {
    minAmount = _minAmount;
    maxAmount = _maxAmount;
    totalSupply = _total;
    hardCap = _cap;
}
```

Owner can set presale start/end time at any moment without limitations.

```
function setPreSaleTime(uint256 _startTime, uint256 _endTime)
    external
    onlyOwner
{
    preSaleStartTime = _startTime;
    preSaleEndTime = _endTime;
}
```

If owner's account is out of $BUB tokens, presalers cannot claim their tokens.

RECOMMENDATIONS FOR

GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

PresaleBub

GOOD PRACTICES FOUND

- ✔ The owner cannot mint new tokens after deployment
- ✔ The owner cannot stop or pause the contract
- ✔ The owner can set a transaction limit, but can’t lower it than 1% of total supply
- ✔ The smart contract utilizes "SafeMath" to prevent overflows

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 150 SUCCESSFUL CLIENTS**
- **MORE THAN 500 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

SPYWOLF.NETWORK

@SPYWOLFNETWORK

@SPYWOLFOFFICIAL

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.